

An Epipolar Line from a Single Pixel

Tavi Halperin Michael Werman
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel

Abstract

We exploit the following observation to directly find epipolar lines. For a pixel p in Image A all pixels corresponding to p in Image B are on the same epipolar line, or equivalently the image of the line spanning A 's center and p is an epipolar line in B .

Computing the epipolar geometry from feature points between cameras with very different viewpoints is often error prone as an object's appearance can vary greatly between images. This paper extends earlier work based on the dynamics of the scene which was successful in these cases.

The algorithms introduced here for finding corresponding epipolar lines accelerate and robustify previous methods for computing the epipolar geometry in dynamic scenes.

1 Introduction

The fundamental matrix is a basic building block of multiple view geometry and its computation is the first step in many vision tasks. The computation is usually based on pairs of corresponding points. Matching points across images is error prone, especially between cameras with very different viewpoints, and many subsets of points need to be sampled until a good solution is found. In this paper, we address the problem of robustly estimating the fundamental matrix from line correspondences in dynamic scenes.

The fundamental matrix is a 3×3 homogeneous rank two matrix with seven degrees of freedom. The best-known algorithm, for computing the fundamental matrix, is the eight point algorithm by Longuet-Higgins [11]

which was made practical by Hartley [5]. The overall method is based on normalization of the data, solving a set of linear equations and enforcing the rank 2 constraint [13]. The requirement of *eight* point correspondences can be relaxed to seven. This results in a cubic equation with one or three real solutions. The estimation from 7 points is very sensitive to noise. These methods are often followed by a non-linear optimization step.

Usually, the first step for calibrating cameras from moving objects is feature tracking, using e.g. deep features [1]. Khan and Shah[8] tracked features on a plane (people viewed from multiple surveillance cameras), and used their trajectories to compute a planar homography between the cameras. They further assumed long videos and the occasional entrance of a single person into the FOV of each camera, from every side. Similarly to them, we assume temporally synchronized cameras. Meingast et al. [14] used the tracks from a multi-target tracking algorithm as features for correspondences. We, as them, use centroids of detected foreground areas as a proxy for an object's location. Theoretically, estimating geometric properties based on fuzzy measurements such as areas resulting from foreground segmentation, or their centroids is error prone. But, as shown by [14], and also by our experiments, this method is robust, and when followed by a global optimization step it is also accurate.

The fundamental matrix can also be computed from three corresponding pairs of epipolar lines [6]. The one-dimensional homography between the lines can be recovered as the epipolar lines in each of the images intersect at the epipoles. The 3 degrees of freedom for the 1D homography together with the 4 degrees of freedom of the epipoles yield the needed 7 parameters.

There are only a few papers using corresponding epipolar lines to compute the epipolar geometry, [2] treats the



Figure 1: A motion barcode b of a line l is a vector in $\{0, 1\}^N$. The value of $b_l(i)$ is "1" when a moving object intersects the line in frame i (black entries) and "0" otherwise (white entries).

case of still images, and [17, 3, 7] are applicable to videos of dynamic scenes.

Sinha and Pollefeys [17] used the silhouette of a single moving object to find corresponding epipolar lines to calibrate a network of cameras. Ben-Artzi et al. [3] accelerated Sinha's method using a similarity measure for epipolar lines. The similarity measure is a generalization of motion barcodes defined in [4, 15].

This line motion barcode was also used in [7] to find corresponding epipolar lines and is the most relevant paper to ours. In that paper, they found corresponding epipolar lines by matching all pairs of lines between the images using the motion barcode. This paper proposes to drastically reduce the search space for matching epipolar lines utilizing pixels which record multiple depths.

The use of corresponding epipolar lines instead of corresponding points stems from; a) the exponent in RANSAC execution time depends on the size of minimal sets needed 3 for epipolar lines as opposed to 7 for points, b) line pairs can be filtered with motion barcodes even in very disparate views where points cannot.

As in previous methods, we assume that cameras are relatively stationary and that moving objects have been extracted using background subtraction.

2 Motion Barcodes

Motion barcodes of lines are used in the case of synchronized stationary cameras viewing a scene with moving objects. Following background subtraction we get a binary video, where "0" represents static background and "1" moving objects.

Given such a video of N binary frames, the motion barcode of a given image line l [3] is a binary vector b_l in $\{0, 1\}^N$. $b_l(i) = 1$ iff a silhouette of a foreground object intersects at least one pixel of line l at the i^{th} frame. An example of a motion barcode is shown in Figure 1.

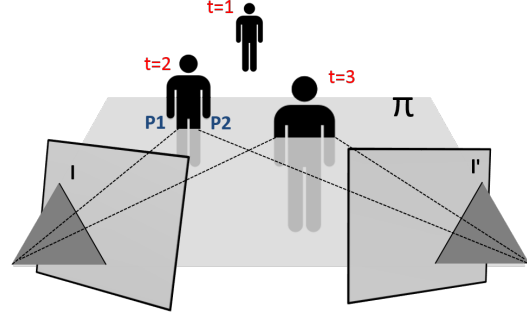


Figure 2: Illustration of a scene with a moving object viewed by two video cameras. The lines l and l' are corresponding epipolar lines, and π is the 3D epipolar plane that projects to l and l' . At time $t = 1$ the object does not intersect the plane π , and thus does not intersect l or l' in the video. At times $t = 2, 3$ the object intersects the plane π , so the projections of this object on the cameras do intersect the epipolar lines l and l' . The motion barcodes of both l and l' is $(0, 1, 1)$

The case of a moving object seen by two cameras is illustrated in Figure 2. If the object intersects the epipolar plane π at frame i , and does not intersect the plane π at frame j , both motion barcodes of lines l and l' will be 1, 0 at frames i, j respectively. Corresponding epipolar lines therefore have highly correlated motion barcodes.

The similarity measure between motion barcodes b and b' is their normalized cross correlation, [4]. To improve reliability, foreground objects are taken to be only some small disc around the centroid of the full computed foreground element.

3 Epipolar Lines

Corresponding epipolar lines are projections of epipolar planes, 3d planes that go through both camera centers. Pixels are projections of 3d rays through a camera center.

The search for corresponding epipolar lines in this paper is based on finding two different correspondences of a pixel. These two correspondences are necessarily on an epipolar line.

The cue to matching, if there is no auxiliary information such as color or reliable shape features, is the

existence/non-existence of movement at time t .

The following notation is used throughout the paper:

p, q, r	pixels
p_A^t	pixel p imaged in Camera A at time t
$q_B \vee r_B$	line between two pixels

Given a pixel p in Image A imaged at two times t and s the corresponding pixels in Image B , q_B^t and r_B^s are on the epipolar line, $q_B \vee r_B$. Likewise, p_A is a point on the corresponding epipolar line to the epipolar line $q_B \vee r_B$.

The algorithm to find corresponding epipolar lines thus has two main steps, (i) finding (at least) two different pixels in B corresponding to a single pixel p in A which results in a single epipolar line in B and (ii) finding a corresponding epipolar line in A from the pencil of lines through p which gives a corresponding pair of epipolar lines.

4 Algorithm

Our algorithm assumes background subtraction. The only pixels we use are the centers of mass of the detected objects.

4.1 Point to Line

For pixels p_A from Camera A , which occur in frames t_1, t_2 we take all the pixels from frames t_1 and t_2 from Camera B , $\{q_B^{t_1}, r_B^{t_1} \dots\}, \{u_B^{t_2}, v_B^{t_2} \dots\}$. Each pair of pixels gives an epipolar line candidate, $L = \{q_B \vee u_B, q_B \vee v_B, \dots, r_B \vee u_B, \dots\}$, Figure 3(b).

For each of the resulting lines, $l \in L$, we find a third pixel $q_B^{t_3}$ on l , such points usually exist in real videos. Let Λ be the lines in Camera A between p_A and all the pixels in Camera A at time t_3 , Figure 3(c). The $\lambda \in \Lambda$ whose motion barcode has the highest normalized cross-correlation to l 's barcode is chosen as l 's partner and the partners with high normalized cross-correlation are considered possible corresponding epipolar lines.

This basic building block is not symmetric with respect to the two cameras. Its result, a pair of candidate epipolar lines, is symmetric. We need to perform this step at least twice in order to proceed, and the cameras may or may not switch roles each time.

When there is enough motion in the scene, using pixels in A that have more than 2 correspondences in B produce

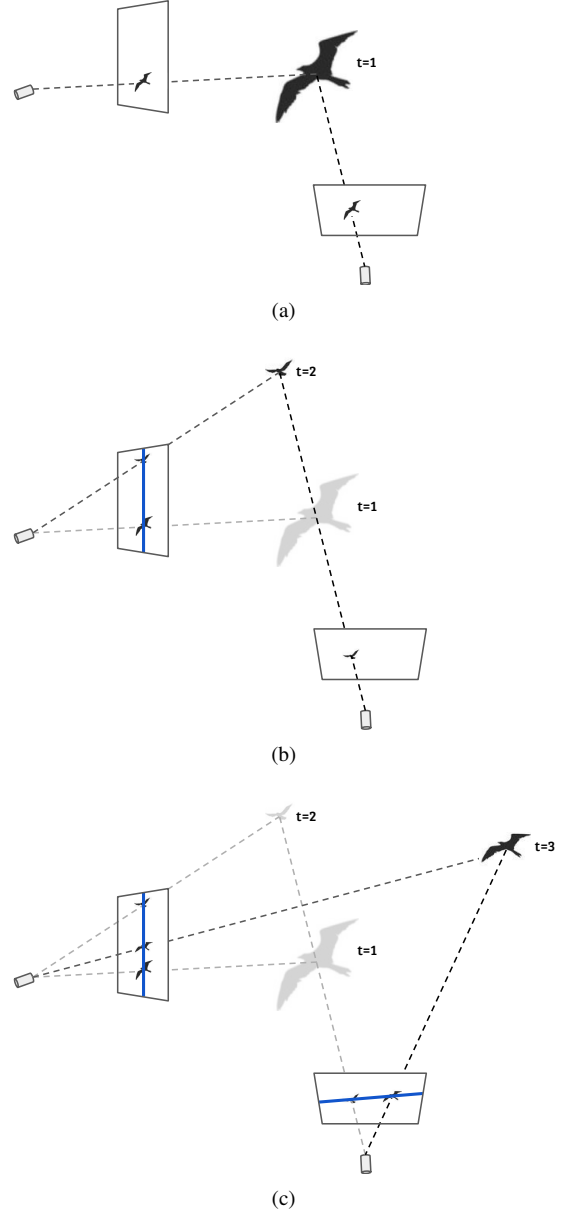


Figure 3: Basic building blocks of our algorithm. (a) Co temporality is the main feature used. (b) The correspondences of a single pixel lie on an epipolar line. (c) Matching epipolar lines have similar motion barcodes.

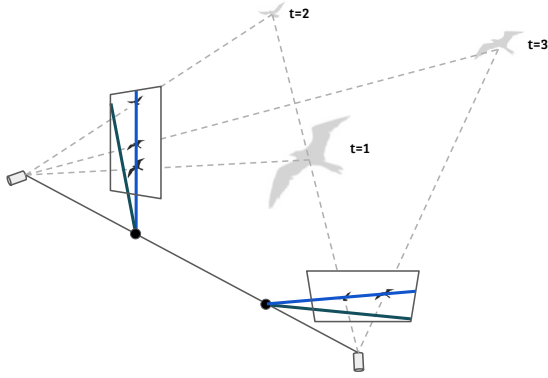


Figure 4: Recovering epipoles from two pairs of epipolar lines.

even better matches with less false positives as it is also checkable if the 3 correspondences in B are co-linear.

4.2 Third Line

We use RANSAC to estimate the location of the epipoles. We sample two pairs of putative corresponding epipolar lines from the previous step, with the probability to sample a pair proportional to its matching score. The intersection of the two epipolar lines suggest epipole locations, e_A and e_B (Figure 4). In order to compute the 1D line homography, a third pair of lines is required. If a third pair of lines is available to us we skip the following step. We pick a random frame t and connect all foreground objects to the epipoles with lines $T_A = \{p_A^t \vee e_A\}$, and $T_B = \{q_B^t \vee e_B\}$, the third correspondence is found by matching the barcodes of lines from T_A and T_B .

These three pairs determine the 1D line homography, which together with the epipoles is sufficient to compute the fundamental matrix.

4.3 Validation

The validation step is carried out in each RANSAC iteration, to evaluate the quality of the estimated epipoles and the homography. Similarly to [2], we compute the 1D line homography between the 3 pairs of lines, sample uniformly 10 lines from the pencil around e_A , transform them to the pencil around e_B , and compute the barcode

correlation between the 10 pairs of lines. The epipole and homography with the highest score are used to compute the fundamental matrix between the cameras. The recovered parameters may be iteratively optimized via bundle adjustment.

4.4 Planar motion

Our algorithm does not work on pure planar motion since it requires two points with different depths on a ray from the camera. However, in the special configuration of one camera on the plane and the other off it, the location of the epipole in the off-plane camera frame may be recovered. In this variant of the algorithm, the first half is the same, with the on-plane camera playing the role of Camera A , with point p . Then, given candidate lines Λ_p from Camera B , we exploit the following facts, (i) there is no motion outside the plane, and (ii) Camera B is off the plane. So that all the motion visible on the epipolar line through p in A is concentrated around p . We then sample p 's barcode from a disc around it, instead of sampling from a line, and use NCC with the barcodes of Λ_p . The one with highest score is kept. We only recover epipolar lines in B , which is not enough to run the validation step, to choose the correct epipole among all intersections of lines. Instead, we ignore lines with matching scores under a certain threshold, and vote for the epipole by maximal consensus voting. This step is carried out using RANSAC, where two lines are drawn in every iteration, their intersection yields the candidate epipole, and the number of lines which agree with the epipole is counted. The candidate with the maximal set of inliers is chosen as the epipole. One definition for inliers is the one used in [7], but a simpler approach which works well when the epipole is inside the image boundaries is measuring whether the perpendicular distance between the epipole and a line is below a certain threshold. As a side effect, this process allows Camera A to be wide-angle with extreme lens distortion. We are not interested in image lines (in A), thus we are not worried about lens distortion, because from the point of view of Camera A we are only interested in rays through pixels, and those are not altered by lens distortion. This even improves accuracy, with more possible epipolar lines (in B) and larger angles between them.

4.5 Static objects

In some cases, features of multiple objects projected to the same point may be extracted. A dynamic object can occlude a static one, for which a different kind of feature (e.g. SIFT[12]) can be detected. The scene can even be fully static with multiple objects detected at the same image point, such as semi transparent surfaces. Various algorithms exist to separate reflections from transmitted light (for example [9, 10, 16]). Two features extracted from the separated layers matched to their corresponding points in the other camera, will produce an epipolar line.

4.6 Coupling with other features

Other features can be used in addition to motion barcodes to guide the search. For example, two objects imaged on p having certain colors (identifiable from other viewing points) will constrain the search in B for objects with matching colors. More complex features such as deep features could be used, for example in a natural scene with high number of moving objects, we can isolate one kind of moving object, e.g. butterflies, and process only their locations.

5 Experiments

We evaluated our algorithm on real and simulated video streams. Since this approach is novel, there are no existing suitable real datasets.

The authors of [7] provided us with their synthetic datasets *cubes* and *thin cubes*. We adopted their area measure and used the same threshold for the definition of inliers.

The main difference is in the first step of the algorithm where we need to find putative corresponding epipolar line pairs. In our case we need to compute the normalized cross correlation between about 10,000 pairs of motion barcodes and in [7] they needed to compute the normalized cross correlation between about 100,000,000 pairs of motion barcodes. The number of barcode correlations which were calculated is four orders of magnitude less and the inlier fraction was reasonable (49.7% in *thin cubes* and 58.1% in *cubes*, on average). With our relatively tiny

number of candidates, The whole algorithm took a small fraction of a second as opposed to minutes.

5.1 Examples

To validate our method on real video examples we filmed several scenes with various types of motion.

Figure 5 shows an example from a real video with planar motion. A wide angle Camera A (GoPro Hero 3+) is mounted at a height of about a meter above the ground facing towards a busy square (right image). Another camera, B , captured the same scene from a typical surveillance angle from a nearby roof (left image).

An example of images of a static scene with a semi transparent surface is shown in Figure 10. Behind the flat window part of a corridor with two doors and a painting on the wall is visible. The reflection on the glass consists of the two cameras with tripods, and buildings behind. The difference in colors between the cameras is due to different white balance. The two red dots marked on the left image (A) are points where two corner points were detected on different surfaces (one behind the glass and one reflected on it), the two layers have been separated and shown individually. The two black boxes show a detected corner point on a door and a point on the tripod of Camera A . The same points are marked with red dots on the right image (B). Since the reflecting surface is flat, the virtual location of the reflected tripod is the same for A and B . Thus, its projection on B must lie on the same epipolar line as the corner of the door. A second line is obtained by applying the same to a second point, and their intersection yields the epipole. For visualization, the reflections of the two camera centers, which of course share an epipolar line, have also been marked and connected by a line.

Representative samples from other real video experiments are shown in Figures 6, 7, 8, and 9.

6 Conclusion

We introduced a method for finding corresponding epipolar lines from multiple correspondences to a single pixel. We conducted experiments with real and synthetic videos, where our method was shown to calibrate cameras similarly to the state of the art but with much less computation.



Figure 5: An example from the Square sequence. (a) An image from the off-plane camera (B), with recovered epipolar lines overlaid. The small box zooms-in on the cameraman of the on-plane camera (A). (b) A frame from the on-plane camera wide angle camera, taken at the same time. The area around the other camera is enlarged for convenience.

Acknowledgement: This research was supported by the Israel Ministry of Science, by the Israel Science Foundation, and by the DFG.

References

- [1] G. Amato, F. Falchi, C. Gennaro, and F. Rabitti. *Similarity Search and Applications: 9th International Conference, SISAP*, chapter YFCC100M-HNfc6: A Large-Scale Deep Features Benchmark for Similarity Search, pages 196–209. Springer International Publishing, Cham, 2016. 1
- [2] G. Ben-Artzi, T. Halperin, M. Werman, and S. Peleg. Epipolar geometry based on line similarity. In *ICPR*, 2016. 1, 4
- [3] G. Ben-Artzi, Y. Kasten, S. Peleg, and M. Werman. Camera calibration from dynamic silhouettes using motion barcodes. In *CVPR’16*, 2016. 2
- [4] G. Ben-Artzi, M. Werman, and S. Peleg. Event retrieval using motion barcodes. In *ICIP’15*, pages 2621–2625, 2015. 2
- [5] R. Hartley. In defense of the eight-point algorithm. *IEEE Trans. PAMI*, 19(6):580–593, 1997. 1
- [6] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1
- [7] Y. Kasten, G. Ben-Artzi, S. Peleg, and M. Werman. Fundamental matrices from moving objects using line motion barcodes. In *European Conference on Computer Vision*, pages 220–228. Springer International Publishing, 2016. 2, 4, 5
- [8] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1355–1360, 2003. 1
- [9] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9), 2007. 5
- [10] Y. Li and M. S. Brown. Single image layer separation using relative smoothness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2752–2759, 2014. 5
- [11] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981. 1
- [12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 5
- [13] Q.-T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *IJCV*, 17(1):43–75, 1996. 1
- [14] M. Meingast, S. Oh, and S. Sastry. Automatic camera network localization using object image tracks. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 1
- [15] D. Pundik and Y. Moses. Video synchronization using temporal signals from epipolar lines. In *ECCV’10*, pages 15–28. Springer, 2010. 2
- [16] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. In *Proceedings of the*

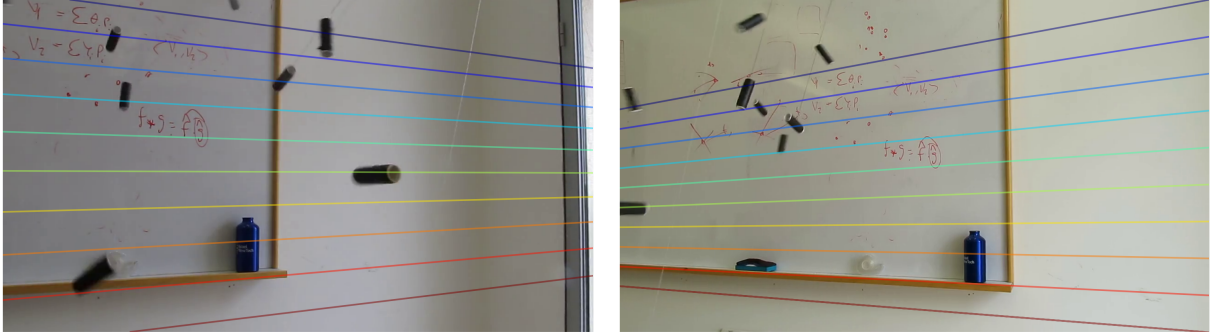


Figure 6: A pair of representing frames from the Threads sequence. Recovered pairs of epipolar lines share the same color. Note that although part of the background is visible in both videos, the epipoles cannot be recovered using only corresponding points from the background, since it's essentially planar.

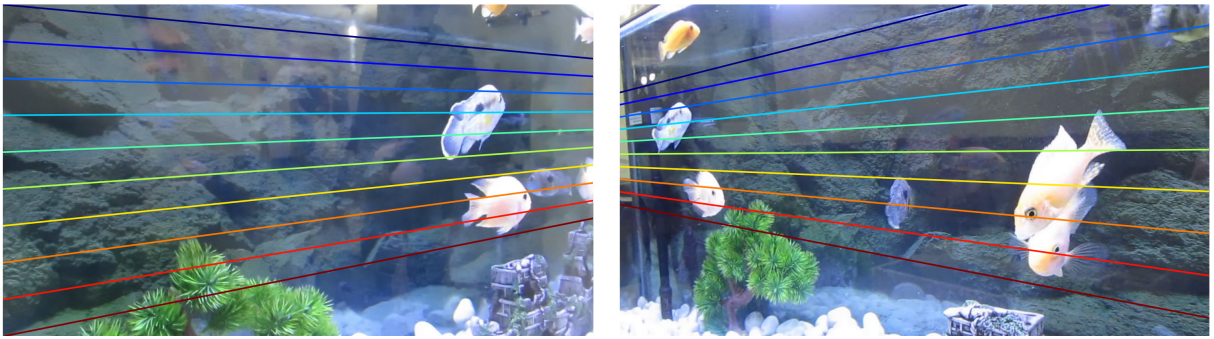


Figure 7: A pair of representative frames from the Fish sequence with overlaying corresponding epipolar lines. Notice that the camera visible in each of the images is not the other camera, but its reflection on the aquarium wall. The two camera reflections are located on corresponding epipolar lines (turquoise). Best viewed in color.

IEEE Conference on Computer Vision and Pattern Recognition, pages 3193–3201, 2015. 5

- [17] S. Sinha and M. Pollefeys. Camera network calibration and synchronization from silhouettes in archived video. *IJCV*, 87(3):266–283, 2010. 2

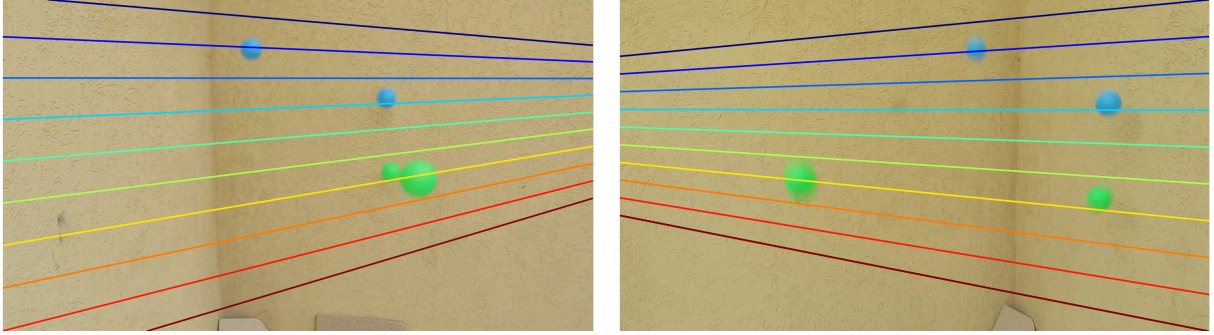


Figure 8: A representative pair of frames from the Balls sequence. When an object is a perfect sphere its 3D centroid projects exactly to the center-of-mass of the detected silhouette (up to the precision of the foreground detection).

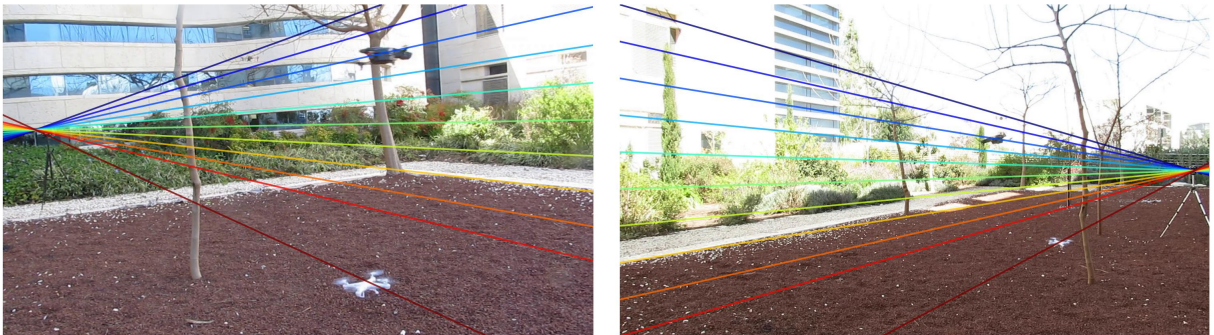
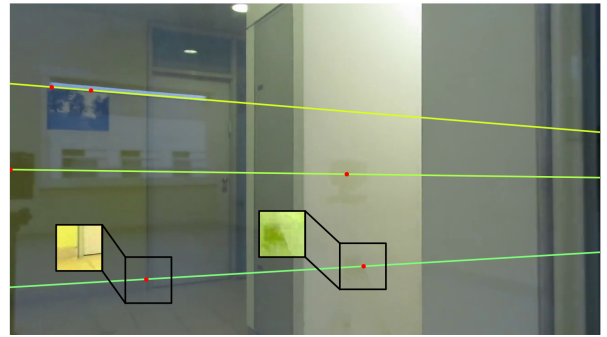


Figure 9: A representative pair of frames from the Drones sequence. Each camera is visible in the other's field of view.



(a)



(b)

Figure 10: In still images of semi transparent surfaces such as windows, multiple objects may be visible at the same image location. (a) Separating the reflections from the transmitted light results in two images (highlighted black boxes), features extracted from these images will correspond to (different) points on an epipolar line in the right image. (b) The two corresponding epipolar lines are shown, and a third one, namely the line connecting the reflections of both camera centers.